

DSTFView: Multi-View Cloud-Edge Workload Forecasting with Dual-Input Spatio-Temporal-Frequency Modeling

Qingzhong Li¹, Hui Ma¹(✉), Yajun Zhang¹, Qingchang Ma¹, Zhou Long¹

¹Xinjiang Key Laboratory of Intelligent Computing and Smart Applications, School
of Software, Xinjiang University, Urumqi, 830017, China

liqingzhong@stu.xju.edu.cn, {huima, zyj}@xju.edu.cn, {maqingchang,
longzhou}@stu.xju.edu.cn

Abstract. With the widespread deployment of edge-side AI inference, edge platforms are increasingly required to support latency-sensitive, highly concurrent, and reliability-critical applications. However, existing methods often struggle to balance multidimensional feature modeling and forecasting efficiency in collaborative cloud-edge environments. To address this issue, we propose DSTFView, a dual-input spatio-temporal-frequency multi-view workload forecasting framework for collaborative cloud-edge environments. It jointly models closeness and period dependencies and extracts spatial, temporal, and frequency-domain dependencies. Besides, it designs an adaptive fusion mechanism and adjusts the contribution of each view to capture abrupt changes. Experimental results on the CPU and TP datasets demonstrate that DSTFView consistently outperforms representative baselines across multiple forecasting horizons and evaluation metrics.

Keywords: cloud-edge collaboration · Workload forecasting · Multi-step prediction · Spatio-temporal-frequency modeling · Adaptive fusion.

1 Introduction

With the ongoing evolution of edge-cloud infrastructures and the rapid expansion of edge-side AI inference services, edge platforms are increasingly expected to accommodate latency-sensitive and highly concurrent applications despite limited resource availability [16, 28]. Under such circumstances, edge workload forecasting is important for optimizing resource management, decreasing resource waste and operational expenses while preserving service quality [3].

However, edge workload forecasting remains a challenging task because it is difficult to uniformly characterize and integrate the heterogeneous, multi-source, and multi-pattern features in edge workloads [14]. Specifically, edge workloads typically exhibit spatial dependencies across sites or virtual machines, temporal dynamics such as burstiness and trend shifts, and frequency-domain regularities arising from service cycles and recurring usage behaviors. Meanwhile, edge

environments also involve multi-relational and time-varying interactions, including geographic proximity, service affinity, and shared-resource coupling. However, due to dynamic deployment, user mobility, and other latent factors, these dependency relationships are often only partially observable and inherently uncertain. As a result, existing methods that focus on temporal dependencies [1, 13, 20, 22], frequency-aware patterns [2], or spatio-temporal dependencies [19] from a single or local perspective are difficult to directly apply to edge workload forecasting. This issue becomes particularly critical in multi-step forecasting, where the lack of unified modeling for heterogeneous relations and multi-dimensional features can easily lead to poor prediction performance.

Another important challenge lies in balancing prediction accuracy and system efficiency in collaborative cloud-edge forecasting [10]. Methods that rely only on edge-side information may overlook cross-site dependencies, whereas cloud-centric solutions often introduce noticeable uplink delay and communication cost. To alleviate this issue, collaborative cloud-edge paradigms have been investigated by combining cloud-side global modeling with edge-side local refinement [11]. However, existing cloud-edge frameworks perform poorly in multi-step forecasting, especially when the exchanged representations cannot adequately capture multi-scale periodic behaviors, or when heterogeneous relations bring in task-irrelevant or unreliable connections that weaken correlation modeling over extended horizons.

To address the aforementioned challenges, we propose DSTFView, an edge workload forecasting framework for collaborative cloud-edge architectures. DSTFView follows a two-stage collaborative cloud-edge forecasting pipeline and introduces a dual-input spatio-temporal-frequency modeling mechanism into both the global and local stages, where the closeness input captures recent dynamics and the period input models recurring temporal patterns. At the cloud-side global stage, the framework leverages lightweight aggregated signals to learn site-level global priors, thereby exploring cross-site dependencies while controlling communication overhead. At the edge-side local stage, it further integrates fine-grained workload sequences with cloud-provided global guidance to refine VM-level multi-step forecasts. In addition, DSTFView incorporates an adaptive multi-view graph fusion mechanism into both stages, enabling the model to dynamically integrate different relational views and enhance dependency modeling.

The main contributions of this work are summarized as follows. *Firstly*, we propose a novel dual-input spatio-temporal-frequency architecture for collaborative cloud-edge multi-step forecasting. *Secondly*, we design an adaptive multi-view graph fusion strategy to dynamically integrate different relational views under varying workload patterns. *Thirdly*, we build a global-local collaborative forecasting pipeline and conduct extensive experiments on the CPU and TP datasets. The results demonstrate that DSTFView consistently outperforms baseline methods.

2 Related Work

2.1 Time Series Prediction

Recently, cloud-side time series forecasting can generally be categorized into three main approaches: temporal, spatio-temporal, and frequency-aware forecasting methods [9].

Temporal forecasting methods. Nie et al. [15] introduced PatchTST, which divides time series into patches and learns temporal dependencies with a Transformer-based architecture. Wu et al. [20] proposed TimesNet, which explicitly models multi-period patterns in long-range forecasting. Liu et al. [13] developed iTransformer by reformulating the attention mechanism to capture variable-wise interactions, whereas Zhang et al. [22] presented Crossformer to model cross-scale dependencies through hierarchical feature interactions.

Spatio-temporal forecasting methods. Xie et al. [8] proposed MegaCRN, which incorporates memory-based components to capture dynamic interactions among nodes. Zhou et al. [26] developed a graph-transformer-based forecasting model that injects spatio-temporal structural information into the attention mechanism and employs specialized modules to characterize heterogeneous traffic dynamics. Zheng and Xie [24] introduced DST-SGNN, which combines Stiefel-manifold-constrained spectral graph operations with an efficient dynamic graph optimization strategy to learn time-varying spatial relationships.

Frequency-aware forecasting methods. They emphasize spectral representations for capturing periodicity and multi-scale patterns. Zhou et al. [27] proposed FEDformer, which utilizes frequency-domain operators to extract informative spectral components for long-horizon prediction. Zhao et al. [23] introduced TFEGRU, which incorporates a time-frequency enhancement module and combines GRU with multi-head self-attention for cloud workload forecasting.

However, existing methods cannot effectively model heterogeneous multi-source relations and spatio-temporal-frequency features, which leads to poor performance in edge workload forecasting.

2.2 Collaborative Cloud-Edge Forecasting

Collaborative forecasting in cloud-edge environments seeks to balance prediction accuracy and system cost by integrating cloud-level global modeling with edge-level local adaptation. Li et al. [11] proposed ELASTIC, a two-stage framework that aggregates VM-level signals at the edge for cloud-side global modeling and then refines predictions through edge-side disaggregation. Also, they [12] extended this idea with improved aggregation/disaggregation modules and extra training objectives for stronger cloud-edge collaboration.

However, existing methods often capture only limited inter-site relations or use coarse fusion strategies, which makes it difficult to model the complex dependencies in edge environments, particularly when workload periodicity is strong and site correlations are diverse.

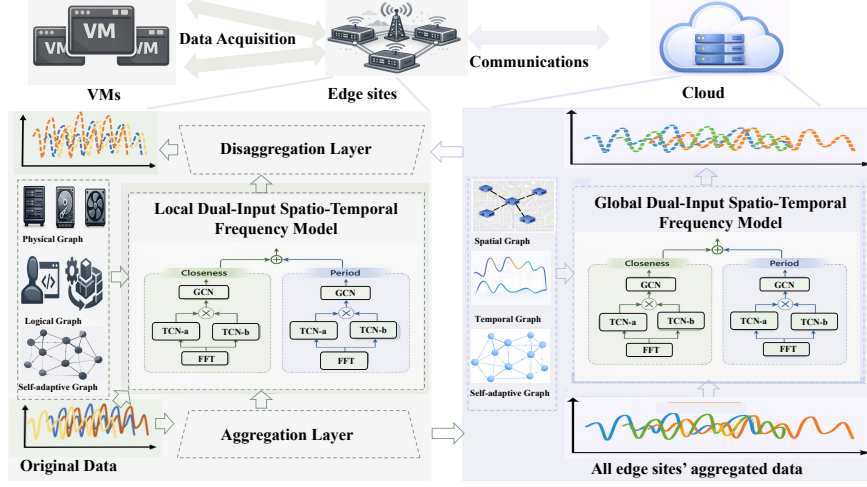


Fig. 1. Structure of DSTFView model.

3 Method

3.1 Framework Overview

As shown in Fig. 1, DSTFVIEW adopts a collaborative two-stage cloud-edge forecasting framework. At the edge side, historical VM-level workload traces are collected at each site and compressed by a learnable *Aggregation Layer* into lightweight site-level sequences. These aggregated representations are then transmitted to the cloud to reduce communication cost. In the cloud-side global stage, the uploaded sequences from all participating sites are used to perform global dual-input spatio-temporal-frequency (STF) modeling over multi-view site graphs, producing coarse-grained site-level multi-step forecasts. The cloud-side predictions are subsequently returned to each edge site, where a local model carries out dual-input STF modeling at VM granularity based on intra-site multi-view graphs. Finally, a *Disaggregation Layer* projects the site-level cloud forecasts back to the VM level and combines them with local predictions, yielding the final VM-level multi-step forecasting results.

3.2 Notations and Problem Formulation

We consider a collaborative cloud-edge paradigm with M edge sites, where site m hosts N_m VMs. Let C denote the number of workload variables and H the prediction horizon. For dual-input temporal modeling, let P_c and P_p denote the lengths of the closeness sequence and the period sequence, respectively.

At time t , the VM-level closeness input and period input at site m are denoted by $\mathbf{X}_m^c \in \mathbb{R}^{N_m \times P_c \times C}$ and $\mathbf{X}_m^p \in \mathbb{R}^{N_m \times P_p \times C}$, respectively. The prediction

target is denoted by $\mathbf{X}_m^{t:t+H-1} \in \mathbb{R}^{N_m \times H \times C}$. Our goal is to learn a collaborative forecaster $\mathcal{F}$ such that

$$\hat{\mathbf{X}}_m^{t:t+H-1} = \mathcal{F}(\mathbf{X}_m^c, \mathbf{X}_m^p), \quad \forall m \in \{1, \dots, M\}. \quad (1)$$

Notation convention. Scalars are denoted by lowercase symbols (e.g., P_c, P_p, H, C); vectors by bold lowercase symbols; matrices/tensors by bold uppercase symbols (e.g., $\mathbf{X}, \mathbf{A}$); and functions/operators by symbols (e.g., $\mathcal{F}, f_{\text{dis}}, \phi$).

3.3 Global Stage: Aggregation and Cloud-side STF Forecasting

The global stage consists of an aggregation layer deployed at each edge site and a cloud-side dual-input STF predictor. Its goal is to compress VM-level traces into lightweight site-level representations and capture cross-site dependencies in the cloud.

For each site m , the VM-level closeness and period inputs are separately compressed into site-level representations through a learnable aggregation layer, yielding $\mathbf{Y}_m^c$ and $\mathbf{Y}_m^p$, respectively, where $\mathbf{Y}_m^c \in \mathbb{R}^{1 \times P_c \times C}$ and $\mathbf{Y}_m^p \in \mathbb{R}^{1 \times P_p \times C}$. By stacking the outputs of all sites, we obtain the cloud-side dual inputs $\mathbf{Y}^c \in \mathbb{R}^{M \times P_c \times C}$ and $\mathbf{Y}^p \in \mathbb{R}^{M \times P_p \times C}$.

On the cloud side, we construct an inter-site multi-view graph set, denoted as $\mathcal{G}_g = [\mathbf{A}_g^{(s)}, \mathbf{A}_g^{(t)}, \mathbf{A}_g^{(a)}]$, where $\mathbf{A}_g^{(s)}$, $\mathbf{A}_g^{(t)}$, and $\mathbf{A}_g^{(a)}$ denote the spatial graph, temporal graph, and Self-adaptive graph, respectively.

The global predictor contains a closeness branch and a period branch, which are used to model recent dynamics and recurring patterns, respectively. For each input branch $r \in \{c, p\}$, FFT is first employed to extract frequency-domain features:

$$\mathbf{F}_g^r = \text{FFT}(\mathbf{Y}^r). \quad (2)$$

Then, the frequency-aware representation is fed into the temporal modeling process, where a pair of temporal convolution branches is used to capture temporal dependencies:

$$\mathbf{T}_g^r = \tanh(\text{TCN}_a(\mathbf{F}_g^r)) \odot \sigma(\text{TCN}_b(\mathbf{F}_g^r)), \quad (3)$$

where $\odot$ denotes element-wise multiplication.

In the structural modeling stage, graph convolutions are performed over the three inter-site graph views, yielding view-specific representations:

$$\mathbf{Z}_g^{r,(k)} = \text{GCN}(\mathbf{A}_g^{(k)}, \mathbf{T}_g^r), \quad k \in \{s, t, a\}. \quad (4)$$

To fully exploit the complementary information among different graph views, we introduce an Adaptive Multi-view Fusion (AMVF) module to adaptively reweight the graph representations from the three views. Specifically, the importance score of the k -th view is computed as

$$e_g^{r,(k)} = \phi_g(\mathbf{Z}_g^{r,(k)}), \quad (5)$$

where $\phi_g(\cdot)$ denotes a learnable scoring function. The fusion weights are then normalized by a Softmax function:

$$\alpha_g^{r,(k)} = \frac{\exp(e_g^{r,(k)})}{\sum_{j \in \{s,t,a\}} \exp(e_g^{r,(j)})}. \quad (6)$$

Accordingly, the fused representation for branch r is given by

$$\tilde{\mathbf{Z}}_g^r = \sum_{k \in \{s,t,a\}} \alpha_g^{r,(k)} \mathbf{Z}_g^{r,(k)}. \quad (7)$$

After obtaining the fused representations from the closeness and period branches, they are merged and fed into the prediction head to generate the site-level forecasting result:

$$\hat{\mathbf{Y}}^{t:t+H-1} = \text{Head}_g(\tilde{\mathbf{Z}}_g^c + \tilde{\mathbf{Z}}_g^p). \quad (8)$$

Here, $\hat{\mathbf{Y}}^{t:t+H-1} \in \mathbb{R}^{M \times H \times C}$ denotes the site-level forecast over the next H time steps. The forecast corresponding to each edge site is then returned to that site as global guidance.

3.4 Local Stage: Edge-side STF Forecasting and Refinement

After receiving the cloud-side forecast, each edge site performs local STF modeling at the VM level and fuses the local prediction with the returned global guidance to generate the final output.

For each site m , we construct an intra-site multi-view graph set $\mathcal{G}_m = [\mathbf{A}_m^{(p)}, \mathbf{A}_m^{(l)}, \mathbf{A}_m^{(a)}]$, where $\mathbf{A}_m^{(p)}$, $\mathbf{A}_m^{(l)}$, and $\mathbf{A}_m^{(a)}$ denote the physical graph, logical graph, and self-adaptive graph, respectively, which characterize correlations among VMs from different perspectives.

Similar to the global predictor, the local predictor also adopts a dual-input STF architecture. For each input branch $r \in \{c, p\}$, frequency-domain features are first extracted as

$$\mathbf{F}_m^r = \text{FFT}(\mathbf{X}_m^r). \quad (9)$$

Temporal dependencies are then modeled by gated temporal convolutions:

$$\mathbf{T}_m^r = \tanh(\text{TCN}_a(\mathbf{F}_m^r)) \odot \sigma(\text{TCN}_b(\mathbf{F}_m^r)). \quad (10)$$

Next, graph convolutions are conducted over the three intra-site graph views:

$$\mathbf{Z}_m^{r,(k)} = \text{GCN}(\mathbf{A}_m^{(k)}, \mathbf{T}_m^r), \quad k \in \{p, l, a\}. \quad (11)$$

Likewise, the local stage also employs AMVF to adaptively fuse the graph representations under different views:

$$e_m^{r,(k)} = \phi_m(\mathbf{Z}_m^{r,(k)}), \quad (12)$$

Table 1. Prediction performance across multiple horizons on CPU dataset.

Models	H=1			H=4			H=8			H=12		
	MAE	SMAPE	R^2	MAE	SMAPE	R^2	MAE	SMAPE	R^2	MAE	SMAPE	R^2
LSTM	0.166	0.191	0.974	0.235	0.220	0.948	0.325	0.263	0.912	0.405	0.295	0.873
Informer [25]	0.514	0.559	0.830	0.488	0.523	0.845	0.489	0.509	0.851	0.499	0.507	0.848
Autoformer [21]	0.387	0.800	0.910	0.403	0.758	0.901	0.445	0.759	0.885	0.483	0.772	0.870
TimesNet [20]	0.206	0.214	0.962	0.236	0.219	0.948	0.280	0.235	0.934	0.319	0.251	0.921
TimeXer [18]	0.191	0.202	0.900	0.218	0.206	0.884	0.257	0.220	0.869	0.293	0.236	0.865
iTransformer [13]	0.197	0.201	0.915	0.221	0.208	0.884	0.259	0.220	0.882	0.295	0.239	0.856
Crossformer [22]	0.202	0.206	0.936	0.233	0.212	0.929	0.274	0.224	0.917	0.315	0.242	0.897
STGCN	0.179	0.320	0.974	0.242	0.329	0.949	0.327	0.355	0.914	0.406	0.382	0.877
DCRNN	0.222	0.243	0.953	0.269	0.244	0.936	0.352	0.270	0.903	0.430	0.299	0.865
GWNET	0.173	0.216	0.974	0.239	0.238	0.947	0.324	0.268	0.913	0.400	0.298	0.878
ASTGCN [7]	0.414	0.375	0.887	0.423	0.360	0.878	0.462	0.366	0.858	0.507	0.383	0.832
GCRN [17]	0.210	0.356	0.968	0.259	0.347	0.945	0.340	0.363	0.912	0.417	0.380	0.873
HGCN [5]	0.212	0.456	0.971	0.259	0.437	0.948	0.330	0.453	0.918	0.389	0.473	0.889
OGCGRN [6]	0.199	0.369	0.968	0.262	0.366	0.943	0.350	0.397	0.906	0.430	0.403	0.867
OTSGCCN [4]	0.163	0.237	0.975	0.232	0.251	0.948	0.318	0.279	0.914	0.397	0.305	0.877
ELASTIC [11]	0.163	0.189	0.976	0.213	0.202	0.959	0.265	0.221	0.940	0.307	0.236	0.920
XELASTIC [12]	0.159	0.188	0.976	0.202	0.200	0.961	0.247	0.218	0.948	0.279	0.233	0.937
DSTFView	0.147	0.141	0.979	0.192	0.163	0.962	0.232	0.180	0.949	0.262	0.191	0.940

$$\alpha_m^{r,(k)} = \frac{\exp\left(e_m^{r,(k)}\right)}{\sum_{j \in \{p,l,a\}} \exp\left(e_m^{r,(j)}\right)}, \quad (13)$$

$$\tilde{\mathbf{Z}}_m^r = \sum_{k \in \{p,l,a\}} \alpha_m^{r,(k)} \mathbf{Z}_m^{r,(k)}. \quad (14)$$

Accordingly, the local predictor outputs the VM-level local forecast:

$$\hat{\mathbf{X}}_{m,\text{local}}^{t:t+H-1} = \text{Head}_l\left(\tilde{\mathbf{Z}}_m^c + \tilde{\mathbf{Z}}_m^p\right). \quad (15)$$

Since the cloud-side forecast is generated at the site level, it is first mapped back to the VM level through a disaggregation layer, yielding the VM-aligned global guidance:

$$\hat{\mathbf{X}}_{m,\text{global}}^{t:t+H-1} = f_{\text{dis}}\left(\hat{\mathbf{Y}}_m^{t:t+H-1}\right). \quad (16)$$

Finally, the VM-level forecast is obtained by adaptively fusing the local prediction and the disaggregated global guidance:

$$\hat{\mathbf{X}}_m^{t:t+H-1} = \lambda_m \odot \hat{\mathbf{X}}_{m,\text{local}}^{t:t+H-1} + (1 - \lambda_m) \odot \hat{\mathbf{X}}_{m,\text{global}}^{t:t+H-1}, \quad (17)$$

where λ_m is a learnable gating tensor.

4 Experimental Setup

4.1 Datasets

To evaluate the effectiveness of *DSTFView*, we conduct experiments on two real-world datasets collected from Alibaba Cloud-native Edge Node Service (ENS)

Table 2. Performance comparison across multiple horizons on TP dataset.

Models	H=1			H=4			H=8			H=12		
	MAE	SMAPE	R^2	MAE	SMAPE	R^2	MAE	SMAPE	R^2	MAE	SMAPE	R^2
LSTM	19.631	0.963	0.564	20.675	0.937	0.548	23.009	0.971	0.522	25.083	1.004	0.506
Informr [25]	18.256	1.328	0.779	17.870	1.222	0.773	18.234	1.205	0.777	18.698	1.219	0.771
Autoformer [21]	12.380	1.451	0.903	12.714	1.385	0.887	14.008	1.379	0.871	15.232	1.383	0.854
TimesNet [20]	5.764	0.335	0.976	7.018	0.359	0.950	8.822	0.383	0.934	10.326	0.417	0.918
TimeXer [18]	5.357	0.329	0.910	6.571	0.333	0.884	8.115	0.357	0.868	9.489	0.391	0.852
iTransformer [13]	5.551	0.323	0.924	6.775	0.326	0.918	8.439	0.350	0.882	9.853	0.394	0.866
Crossformer [22]	5.695	0.326	0.958	6.879	0.330	0.922	8.723	0.374	0.905	10.127	0.408	0.899
STGCN	6.394	1.375	0.986	8.618	1.329	0.940	12.002	1.333	0.884	14.905	1.367	0.848
DCRNN	33.562	0.514	0.075	32.616	0.568	0.119	34.300	0.612	0.123	35.634	0.826	0.127
GWNET	4.889	0.710	0.981	7.123	0.684	0.945	10.387	0.718	0.899	13.281	0.762	0.853
ASTGCN [7]	13.726	1.167	0.819	13.760	1.091	0.813	14.784	1.095	0.817	15.958	1.109	0.801
GCRN [17]	15.820	1.181	0.733	16.804	1.125	0.726	18.948	1.159	0.700	21.022	1.183	0.674
HGCN [5]	6.427	0.929	0.980	7.761	0.903	0.944	10.075	0.937	0.918	12.199	0.960	0.882
OGCGRN [6]	16.981	1.222	0.734	18.185	1.186	0.718	20.559	1.200	0.682	22.723	1.234	0.656
OTSGCCN [4]	3.845	1.156	0.987	6.539	1.100	0.951	9.903	1.124	0.895	12.726	1.098	0.869
ELASTIC [11]	3.859	0.290	0.981	5.633	0.314	0.965	7.706	0.348	0.949	9.250	0.382	0.933
XELASTIC [12]	3.782	0.314	0.995	5.446	0.318	0.973	7.150	0.352	0.953	8.374	0.385	0.937
DSTFView	3.496	0.237	0.998	5.189	0.251	0.979	6.753	0.295	0.957	7.877	0.319	0.939

and the Network Edge Platform (NEP)¹. These datasets represent typical workload patterns in large-scale cloud-edge environments.

CPU dataset, collected from Alibaba ENS, it contains CPU utilization traces from 1,728 IaaS virtual machines (VMs) across 77 geographically distributed edge sites in China. **TP dataset**, collected from Alibaba NEP, it records average network traffic, including uplink and downlink throughput measured in *Mbps*. Compared with CPU utilization, network throughput is more sensitive to user mobility and bursty service requests.

4.2 Experimental Details

We adopt Z-score standardization and divide the dataset into training, validation, and test subsets in chronological order with a ratio of 7:2:1. All proposed methods are trained with the Adam optimizer, using an initial learning rate of 0.001 and an ExponentialLR scheduler for decay. The loss function is mean squared error (MSE). For both the global and local stages, the batch size is set to 64 and the maximum number of training epochs is 50. All experiments are executed on a Linux server equipped with four NVIDIA TITAN X (Pascal) GPUs, each with 12 GB memory.

5 Experimental Results

5.1 Overall Comparison

Tables 1 and 2 report the forecasting results on the CPU and TP datasets over four prediction horizons (1/4/8/12) in terms of MAE, SMAPE, and R^2 . We

¹ <https://github.com/xumengwei/EdgeWorkloadsTraces>.

Table 3. Ablation study of DSTFView under different global graph configurations.

Datasets	Configurations	H=1		H=4		H=8		H=12	
		MAE	SMAPE	MAE	SMAPE	MAE	SMAPE	MAE	SMAPE
CPU	Identity	2.953	0.050	3.965	0.076	5.208	0.111	6.552	0.145
	spatial-only	2.557	0.050	3.241	0.068	4.174	0.091	5.203	0.114
	temporal-only	2.745	0.050	3.579	0.072	4.599	0.099	5.699	0.126
	adaptive-only	2.409	0.050	2.932	0.064	3.721	0.082	4.608	0.100
	spatial-temporal	2.379	0.049	2.894	0.064	3.664	0.081	4.489	0.098
	tim-spa-adaptive(AMVF)	2.348	0.049	2.826	0.063	3.579	0.079	4.370	0.095
TP	Identity	703.272	0.307	771.352	0.325	932.822	0.367	1152.779	0.416
	spatial-only	622.728	0.280	683.011	0.297	825.989	0.335	1020.755	0.379
	temporal-only	391.445	0.203	429.338	0.215	519.213	0.242	641.642	0.274
	adaptive-only	556.219	0.258	610.064	0.273	737.771	0.308	911.735	0.349
	spatial-temporal	225.359	0.147	247.175	0.156	298.917	0.176	369.401	0.199
	tim-spa-adaptive(AMVF)	214.357	0.143	235.108	0.152	284.324	0.171	351.367	0.194

Table 4. Ablation study of DSTFView under different local graph configurations.

Datasets	Configurations	H=1		H=4		H=8		H=12	
		MAE	SMAPE	MAE	SMAPE	MAE	SMAPE	MAE	SMAPE
CPU	Identity	0.182	0.157	0.238	0.181	0.288	0.200	0.324	0.213
	physical-only	0.959	0.362	1.253	0.416	1.518	0.461	1.711	0.489
	logical-only	0.168	0.151	0.219	0.174	0.265	0.193	0.299	0.204
	adaptive-only	0.154	0.145	0.201	0.167	0.243	0.184	0.274	0.196
	phy-log-adaptive (AMVF)	0.147	0.141	0.192	0.163	0.232	0.180	0.262	0.191
TP	Identity	4.363	0.265	6.478	0.280	8.430	0.329	9.833	0.356
	physical-only	3.805	0.247	5.649	0.262	7.351	0.308	8.575	0.332
	logical-only	3.817	0.248	5.667	0.262	7.375	0.308	8.602	0.333
	adaptive-only	3.641	0.242	5.406	0.256	7.035	0.301	8.206	0.325
	phy-log-adaptive (AMVF)	3.496	0.237	5.189	0.251	6.753	0.295	7.877	0.319

compare *DSTFView* with representative temporal, spatiotemporal, and collaborative cloud-edge baselines.

Temporal baselines, such as iTransformer and TimerXer, model each site or VM independently and therefore cannot explicitly capture inter-entity dependencies. Spatiotemporal methods, such as OGCGRN and OTSGCCN, consider graph relations, but many rely on a single predefined graph and may not fully reflect the heterogeneous correlations in edge workloads. ELASTIC and XELASTIC adopt a collaborative cloud-edge pipeline, but they cannot jointly model temporal and frequency patterns or adaptively exploit multiple relational views.

As shown in Tables 1 and 2, *DSTFView* consistently achieves the best performance on both datasets. This advantage mainly comes from two aspects. Firstly, the proposed dual-input STF modeling captures recent dynamics, periodic patterns, and spatial correlations in a unified manner. Also, the adaptive multi-view fusion mechanism dynamically integrates multiple relational views in both the global and local stages, thereby enabling more effective dependency modeling and improving forecasting accuracy.

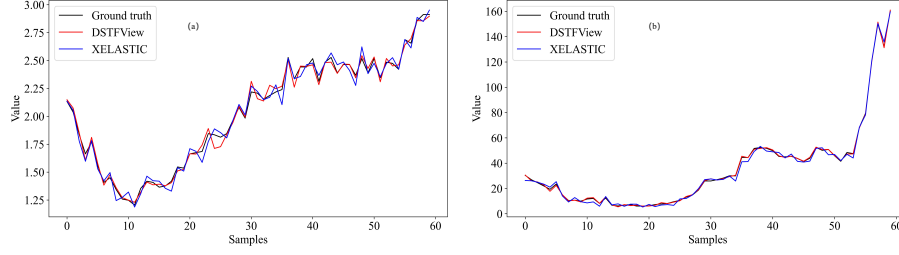


Fig. 2. Visualization results on the CPU and TP dataset.

5.2 Ablation Study

In this section, we conduct an ablation study to evaluate the contributions of its key components at the *Global Stage* and the *Local Stage*.

For the global stage, we examine single-view variants, a dual-view variant (*spatial-temporal*), and the full model integrating temporal, adaptive, and spatial views. As reported in Table 3, multi-view combinations generally outperform single-view settings, and the complete model achieves the best overall results, demonstrating the benefit of jointly modeling temporal, frequency, and spatial information.

For the local stage, we compare different relation configurations, including Identity, physical-only, logical-only, adaptive-only, and the fused *phy-log-adaptive* setting (AMVF). Table 4 shows that adaptive relations consistently improve over the baseline and other single-view settings, while the fused AMVF configuration achieves the best performance, confirming the effectiveness of combining multiple relation views.

5.3 Visualization Analysis

Fig. 2 presents qualitative forecasting examples for a representative VM under the one-step-ahead setting, where subfigure (a) corresponds to the CPU dataset and subfigure (b) corresponds to the TP dataset. In the figure, the black curve indicates the ground-truth series, the red curve denotes *DSTFView*, and the blue curve represents the strongest competing baseline, *XELASTIC*. As can be observed from Fig. 2, *DSTFView* tracks the target series more closely, especially around peaks, valleys, and local trend reversals, whereas *XELASTIC* exhibits larger deviations in these regions. This advantage can be attributed to the proposed model’s ability to jointly capture periodic components and short-term high-frequency fluctuations through adaptive AMVF-based fusion, thereby yielding more accurate predictions on both datasets.

6 Conclusion

In this study, we propose *DSTFView*, a collaborative cloud-edge framework for workload forecasting. It uses dual-input spatio-temporal-frequency multi-view

modeling to jointly capture cross-site dependencies, recent dynamics, and periodic patterns through cloud-side global forecasting and edge-side local forecasting. The adaptive fusion mechanism further highlights informative graph structures. Experimental results on the CPU and TP datasets demonstrate that *DSTFView* consistently outperforms representative baselines across multiple forecasting horizons. Future work will focus on dynamic graph construction, continual learning, and integration with resource management tasks such as auto-scaling and scheduling to improve adaptability and practical value in real edge environments.

References

1. Chen, P., Zhang, Y., Cheng, Y., Shu, Y., Wang, Y., Wen, Q., Yang, B., Guo, C.: Pathformer: Multi-scale transformers with adaptive pathways for time series forecasting. In: International Conference on Learning Representations (2024)
2. Fan, W., Zheng, S., Yi, X., Cao, W., Fu, Y., Bian, J., Liu, T.Y.: DEPTS: Deep expansion learning for periodic time series forecasting. In: International Conference on Learning Representations (2022)
3. Ghahari-Bidgoli, M., Ghobaei-Arani, M., Sharif, A.: An efficient task offloading and auto-scaling approach for IoT applications in edge computing environment. *Computing* **107**(5), 125 (2025)
4. Guo, K., Hu, Y., Qian, Z.S., Sun, Y., Gao, J., Yin, B.: An optimized temporal-spatial gated graph convolution network for traffic forecasting. *IEEE Intelligent Transportation Systems Magazine* **14**(1), 153–162 (2022)
5. Guo, K., Hu, Y., Sun, Y., Qian, S., Gao, J.: Hierarchical graph convolution network for traffic forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 151–159 (2021)
6. Guo, K., et al.: Optimized graph convolution recurrent neural network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems* **22**(2), 1138–1149 (2021)
7. Guo, S., Lin, Y., Feng, N., Song, C.: Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 922–929 (2019)
8. Jiang, R., Wang, X., Jie, Y., Chen, C., Kobayashi, K., Song, X., Fukushima, S., Suzumura, T.: Spatio-temporal meta-graph learning for traffic forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence (2023)
9. Kong, X., Chen, Z., Liu, W., Ning, K., Zhang, L., Marier, S.M., Liu, Y., Chen, Y., Xia, F.: Deep learning for time series forecasting: a survey. *International Journal of Machine Learning and Cybernetics* **16**, 5079–5112 (2025)
10. Kreković, D., Krivić, P., Žarko, I.P., Kušek, M., Le-Phuoc, D.: Reducing communication overhead in the iot-edge-cloud continuum: A survey on protocols and data reduction strategies. *Internet of Things* **31**, 101553 (2025). <https://doi.org/10.1016/j.iot.2025.101553>
11. Li, Y., Yuan, H., Fu, Z., Xu, M., Wang, S.: ELASTIC: Edge workload forecasting based on collaborative cloud-edge deep learning. In: Proceedings of the ACM Web Conference. pp. 3056–3066 (2023)
12. Li, Y., Zhao, P., Ma, X., Yuan, H., Fu, Z., Xu, M., Wang, S.: A collaborative cloud-edge approach for robust edge workload forecasting. *IEEE Transactions on Mobile Computing* **24**(4), 2861–2875 (2025)

13. Liu, Y., Hu, T., Zhang, H., Wu, H., Wang, S., Ma, L., Long, M.: iTransformer: Inverted transformers are effective for time series forecasting. arXiv preprint arXiv:2310.06625 (2023)
14. Ma, X., Xu, M., Li, Q., Li, Y., Zhou, A., Wang, S.: Edge workload prediction based on deep learning. In: 5G Edge Computing: Technologies, Applications and Future Visions, pp. 45–61. Springer, Singapore (2024)
15. Nie, Y., H. Nguyen, N., Sinthong, P., Kalagnanam, J.: A time series is worth 64 words: Long-term forecasting with transformers. In: International Conference on Learning Representations (2023)
16. Ning, Z., Hu, H., Wang, X., Guo, L., Guo, S., Wang, G., Gao, X.: Mobile edge computing and machine learning in the Internet of unmanned aerial vehicles: A survey. ACM Computing Surveys **56**(1), 1–31 (2023)
17. Seo, Y., Defferrard, M., Vandergheynst, P., Bresson, X.: Structured sequence modeling with graph convolutional recurrent networks. In: International Conference on Neural Information Processing. pp. 362–373. Springer (2018)
18. Wang, Y., Wu, H., Dong, J., Qin, G., Zhang, H., Liu, Y., Qiu, Y., Wang, J., Long, M.: TimeXer: Empowering transformers for time series forecasting with exogenous variables. In: Advances in Neural Information Processing Systems. vol. 37, pp. 469–498 (2024)
19. Wang, Z., Hu, R., Zhong, B., Cheng, D., Feng, J., Tan, S.: St-graphormer: spatio-temporal graph transformer for end-to-end traffic forecasting. The Journal of Supercomputing **81**(896) (2025)
20. Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., Long, M.: TimesNet: Temporal 2D-Variation modeling for general time series analysis. In: International Conference on Learning Representations (2023)
21. Wu, H., Xu, J., Wang, J., Long, M.: Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In: Advances in Neural Information Processing Systems. pp. 22419–22430 (2021)
22. Zhang, Y., Yan, J.: Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In: International Conference on Learning Representations (2023)
23. Zhao, F., Lin, W., Lin, S., Zhong, H., Li, K.: TFEGRU: Time-frequency enhanced gated recurrent unit with attention for cloud workload prediction. IEEE Transactions on Services Computing (2024)
24. Zheng, J., Xie, L.: A dynamic stiefel graph neural network for efficient spatio-temporal time series forecasting. In: International Joint Conference on Artificial Intelligence (2025)
25. Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., Zhang, W.: Informer: Beyond efficient transformer for long sequence time-series forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 11106–11115 (2021)
26. Zhou, J., Liu, E., Chen, W., Zhong, S., Liang, Y.: Navigating spatio-temporal heterogeneity: A graph transformer approach for traffic forecasting. arXiv preprint arXiv:2408.10822 (2024)
27. Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., Jin, R.: FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In: Proceedings of the 39th International Conference on Machine Learning. vol. 162, pp. 27268–27286 (2022)
28. Zhu, G., Lyu, Z., Jiao, X., et al.: Pushing AI to wireless network edge: An overview on integrated sensing, communication, and computation towards 6G. Science China Information Sciences **66**(3), 130301 (2023)